REMARKS

In response to the Final Office Action dated September 9, 2004, the Applicant is filing this Response without amendment. At the time of the Office Action, claims 1-52 were pending. No new claims are being added and no claims are being canceled. Accordingly, claims 1-52 remain currently pending.

In the Office Action, claims 1-4, 7-10, 13-17, 20-25, 28-33, 36-40, 43-48, 51 and 52 were rejected under 35 U.S.C. § 103(a) as being obvious over Monier (U.S. Patent No. 5,974,455) in view of Najork (U.S. Patent No. 6,301,614 "Najork '614"). Additionally, claim 22 was rejected under 35 U.S.C. § 103(a) as being obvious over Monier in view of Najork (U.S. Patent No. 6,321,265 "Najork '265") in further view of Najork '614. Claims 5, 6, 11, 12, 18, 19, 26, 27, 31, 32, 34, 35, 49, and 50 were rejected under 35 U.S.C. § 103(a) as being obvious over Monier in view of Najork '614 in further view of Cabrera et al. (U.S. Patent No. 5,953,729). Each of the rejections is addressed in detail below.

First Rejection Under 35 U.S.C. § 103

With respect to the rejection of independent claims 1, 13, 23, 31, 38, and 46 based on Monier in view of Najork '614, the Examiner stated:

> 4. In reference to claims 1, 13, 23, 31, 38 and 46, Monier teaches downloading data sets from among a plurality of host computers comprising the following steps:
>
> Storing representations of data set addresses in a set of data structures, including a first buffer, a second buffer and a first disk file, wherein representations of data set addresses stored in the first disk file are ordered (column 3, lines 1-35, Monier discloses storing URL representations in a set of data structures, including a hash table (stored in random access memory (RAM)), an append buffer (stored in RAM) and a

sequential disk file, wherein the representations are stored sequentially in the disk file.

Selecting as a current buffer one of the first and second buffers (column 6, lines 35-45, Monier discloses selecting and managing a current buffer among the hash table and append buffer).

Downloading at least one data set that includes addresses of one or more referred data sets (column 5, lines 20-30, Monier discloses fetching web pages that include URL's of one or more referred web pages).

Identifying the addresses of the one or more referred data sets (column 5, lines 20-30, Monier discloses analyzing and identifying the addresses of the one or more referred web pages).

For each identified address:

Generating a representation of the identified address (column 5 line 55 – column 6 line 22, Monier discloses generating a fingerprint representation of the specified URL), and

Determining whether the representation is stored in the buffer without determining whether the representation is stored in the first disk file, and when this determination is negative, storing the representation in the buffer (column 5 line 43 – column 6 line 22, Monier discloses determining whether the representation is stored in the hash table, and when this determination is negative, storing the representation in the hash table). Monier inherently teaches without determining whether the representation is stored in the first disk file.

When the buffer reaches a predefined full condition:

Ordering the contents of the buffer according to the representations (column 6, line 1-33, Monier discloses ordering the contents of the hash table according to the fingerprint representations), and

Monier discloses appending contents of the hash table into the contents of the disk file (column 6 line 22 – column 7 line 12). Monier fails to teach performing an ordered merge of the contents of the buffer into the contents of the first disk file wherein the ordered merge comprises preventing duplication of any of the representations of data set addresses stored in the first disk file. However, Najork teaches sorting an index of

representations and performing a sorted merge of the index with a disk file (column 3 line 30 – column 4 line 45 and column 6 line 1 – column 7 line 25).

It would have been obvious for one having ordinary skill in the art to perform a sorted merge of the hash table with the disk file as per the teachings of Najork for facilitating look-up operations on the disk file.

Selecting the other buffer as the current buffer, wherein the previously current buffer is identified as a non-current buffer (column 6, lines 22-67, Monier discloses selecting the append buffer as the current buffer, wherein the hash table is identified as a non-current buffer).

Final Office Action, pages 2-4.

The Applicant respectfully traverses this rejection. The burden of establishing a *prima facie* case of obviousness falls on the Examiner. *Ex parte Wolters and Kuypers*, 214 U.S.P.Q. 735 (P.T.O. Bd. App. 1979). Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teaching or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital*, 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Accordingly, to establish a *prima facie* case, the Examiner must not only show that the combination includes *all* of the claimed elements, but also a convincing line of reason as to why one of ordinary skill in the art would have found the claimed invention to have been obvious in light of the teachings of the references. *Ex parte Clapp*, 227 U.S.P.Q. 972 (Bd. Pat. App. & Inter. 1985). When prior art references require a selected combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gained from the invention itself, i.e., something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination. *Uniroyal Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 5 U.S.P.Q.2d 1434 (Fed. Cir. 1988).

Obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent some teachings or suggestion supporting the combination. *ACS Hospital Systems, Inc. v. Montefiore Hospital,* 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed Cir. 1984). One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention. *In re Fine,* 837 F.2d 1071, 5 U.S.P.Q. 2d 1596 (Fed. Cir. 1988).

It is improper to combine references where the references teach away from their combination. *In re Grasselli,* 713 F.2d 731, 743, 218 U.S.P.Q. 769, 779 (Fed. Cir. 1983); M.P.E.P. § 2145. Moreover, if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious. *In re Ratti,* 270 F.2d 810, 123 U.S.P.Q. 349 (C.C.P.A. 1959); see M.P.E.P. § 2143.01.

Additionally, the Examiner's reliance on a theory of inherency requires that the extrinsic evidence make clear that the missing descriptive matter is *necessarily* present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. *In re Robertson,* 169 F.3d 743, 49 U.S.P.Q.2d 1949 (Fed. Cir. 1999) (Emphasis Added). The mere fact that a certain thing *may* result from a given set of circumstances is not sufficient. *Id.* In relying upon the theory of inherency, the Examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic *necessarily* flows from the teachings of the applied prior art. *Ex parte Levy,* 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Inter. 1990) (emphasis in original). The

Examiner, in presenting the inherency argument, bears the evidentiary burden and must adequately satisfy this burden. *See id.*

In the present case, the combination of Monier and Najork '614 cannot render the Applicant's claims obvious under 35 U.S.C. § 103 because Monier and Najork '614 do not disclose all of the elements recited in the Applicant's claims. As amended, independent claims 1, 13, 23, 31, 38, and 46 recite "determining whether the representation is stored in the buffer without determining whether the representation is stored in the first disk file" and "performing an ordered merge ... wherein the ordered merge comprises preventing duplication of any of the representations of data set addresses stored in the first disk file." These elements are completely missing from both Monier and Najork '614.

The Applicant stresses that Monier fails to teach or suggest determining whether the representation is stored in the buffer *without determining whether the representation is stored in the first disk file*. The Examiner stated that "Monier *inherently* teaches without determining whether the representation is stored in the first disk file." Final Office Action, page 3 (Emphasis Added). However, the Examiner failed to support this assertion of inherency. Accordingly, the Applicant asserts that the Examiner has not met the evidentiary burden of inherency. The Applicant reminds the Examiner that the Examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic *necessarily* flows from the teachings of the applied prior art.

The Applicant further asserts that Monier does not teach or suggest "performing an ordered merge ... wherein the ordered merge comprises preventing duplication of any of the representations of data set addresses stored in the first disk file" as recited in claims 1, 13, 23,

31, 38, and 46. In contrast, Monier appears to disclose appending the contents of a buffer to the end of a disk file. Col. 6, lines 58-63. Thus, Monier does not disclose "an ordered merge," much less an ordered merge "preventing duplication" of the merged contents. Monier instead discloses "an input buffer 134 and an append buffer 136" that cooperate with a "sequential disk file 150." Col. 6, lines 36-63. The input buffer makes all accesses to the disk file and the append buffer stores all the new entries to be added to the sequential disk file. Col. 6, lines 53-63.

The procedure described by Monier comprises "copying a *first batch* of sequential disk entries from the disk file 150 into the input buffer 134," which indicates that only a portion of the disk file is copied. Col. 7, lines 33-65 (Emphasis added). In fact, the input buffer is merely used to "store a sequentially ordered contiguous *portion* of the ... disk file 150." Col. 6, lines 43-46 (Emphasis added). Further, Monier discloses a Web scooter that then scans the input buffer. Col. 6, line 64 – col. 7, line 8. Each time a scan is completed for all entries in the input buffer, "all updates to the entries in the input buffer are *stored back into the Web information disk file 150* and *all entries in the append buffer 136 are appended to the end of the disk file 150*." *Id.* (Emphasis added). Monier thus discloses scanning the input buffer, which merely consists of a first batch or portion of sequential disk entries, with the Web scooter.

Monier further teaches storing entry updates in the input buffer back into the disk file and appending all entries in the append buffer to the end of the disk file. Accordingly, Monier teaches that the remaining portion of the disk file (e.g., other batches) *may be duplicated.* This makes it clear that nowhere does Monier disclose "performing an ordered

merge ... *preventing duplication* of any of the representations of data set addresses stored in the first disk file" as recited in claims 1, 13, 23, 31, 38, and 46. (Emphasis Added).

Najork '614 also fails to disclose an ordered merge preventing duplication of the merged contents. Instead, Najork '614 discloses a processing procedure comprising three lookup steps and a merging process wherein the stored contents of a cache and disk file are combined and reorganized into a sorted order. Col. 6, line 1 – col. 7, line 25. In two of the processing procedure steps described in Najork '614, lookups are performed to see if a numeric representation is already stored in one of two caches. Col. 6, lines 4-24. In the third lookup step, if the numeric representation is in neither of the two caches, a lookup is performed to determine whether it is already stored in the disk file. Col. 6, lines 24-36. If the numeric value is not stored in either the caches or the disk file, it is added to one of the caches. Col. 6, lines 53-55. If the cache receiving the numeric value is full, its contents are merged with the disk file. Col. 6, line 63 – col. 7, line 6.

Accordingly, if a group of numeric representations are new to both the caches and the disk file, the disk file may be accessed multiple times to determine its contents. When the Najork '614 procedure activates the merging process, the disk file has *already* been checked for duplicates. Unlike embodiments of the present invention which increase efficiency by performing an "ordered merge ... wherein the ordered merge *comprises* preventing duplication of any of the representations of data set addresses stored in the first disk file," Najork '614 discloses accessing the disk file multiple times to prevent duplication *before* merging the cache with the disk file. (Emphasis Added). In the Final Office Action, the Examiner stated:

> Applicant argues that Najork '614 teaches preventing duplication before merging the cache, which is unlike embodiments of the present invention. However, Najork '614 teaches that the merging process involves the step of preventing duplication of any of the representations of data set addresses stored in the first disk file (column 3 line 30 – column 4 line 45 and column 6 line 1 – column 7 line 25). The claim language does not indicate that the step of preventing duplication occurs after merging, and therefore it is irrelevant as to when the prevention takes place.

Final Office Action, page 13. Again, as discussed above, the Applicant stresses that Najork '614 teaches accessing the disk file multiple times to prevent duplication *before* merging the cache with the disk file. Najork '614 does not teach an ordered merge that *comprises* preventing duplication of any of the representations of data set addresses stored in the first disk file, as recited in the present claims. Accordingly, the Applicant asserts that it is not necessary to modify the claim language to say preventing duplication *after* merging.

In addition, the Applicant asserts that the Examiner's proposed combination of Monier and Najork '614 is improper. Monier teaches that sequential ordering of entries (e.g., appending new entries to the end of a disk file) eliminates latency caused by disk access. Accordingly, Monier discloses a "disk file 150 that is *sequentially* accessed." Col. 5, lines 1-3. In fact, Monier teaches that the input and append buffers access a Web page directory "only in sequential order." Col. 9, lines 56-63. Additionally, Monier discloses adding new entries by appending the contents of an append buffer to the end of the sequential disk file. Col. 6, lines 58-63.

In contrast, Najork '614 does not embrace the desirability of sequential access. Najork '614, in fact, teaches that disk read operations are reduced and efficiency increased by using an index of numerical representations "in *sorted order*." Col. 3, line 49 – col. 4, line

16. Because Monier teaches that sequential access is a primary mechanism to overcome speed limitations and thus to increase efficiency, one of ordinary skill in the art would not be motivated to combine Monier with Najork '614, which describes sorting and ordering numerical representations. Any proposed combination of these references could only be achieved using hindsight gained from the present invention.

In view of the arguments presented above, the Applicant asserts that independent claims 1, 13, 23, 31, 38, and 46 are in condition for allowance. Accordingly, the Applicant requests that the Examiner withdraw the objection of these claims and provide an indication of allowance. Additionally, the Applicant asserts that all claims depending from independent claims 1, 13, 23, 31, 38, and 46 are in condition for allowance based on their respective dependencies and for unique matter recited therein.

Second Rejection Under 35 U.S.C. § 103(a)

With respect to the rejection of claim 22 based on anticipation by Monier in view of Najork '265, the Examiner stated:

> 11. Claim 22 rejected under 35 U.S.C. 103(a) as being obvious over Monier (U.S. Patent No. 5, 974,455) in view of Najork (U.S. Patent No. 6,321,265) in further view of Najork (U.S. Patent No. 6,301,614).
>
> ...
>
> 12. Monier teaches downloading data sets from among a plurality of host computers comprising the following steps:
>
> Storing representations of data set addresses in a set of data structures, including a first buffer, a second buffer and a first disk file, wherein representations of data set addresses stored in the first disk file are ordered (column 3, lines 1-35, Monier discloses storing URL representations in a set of data structures, including a hash table (stored in random access

memory (RAM)), an append buffer (stored in RAM) and a sequential disk file, wherein the representations are stored sequentially in the disk file).

Selecting as a current buffer one of the first and second buffers (column 6, lines 35-45, Monier discloses selecting and managing a current buffer among the hash table and append buffer).

Downloading at least one data set that includes addresses of one or more referred data sets (column 5, lines 20-30, Monier, discloses fetching web pages that include URL's of one or more referred web pages).

Identifying the addresses of the one or more referred data sets (column 5, lines 20-30, Monier discloses analyzing and identifying the addresses of the one or more referred web pages).

Generating a representation of the identified address (column 5 line 55 – column 6 line 22, Monier discloses generating a fingerprint representation of the specified URL), and

Monier discloses determining whether the representation is stored in the hash table (column 5 line 43 – column 6 line 22). Monier fails to teach whether the disk file is empty, and when the representation is not stored in the buffer and the disk file is empty, scheduling the corresponding data set for downloading. However, Najork '265 teaches determining if a queue is empty and if it is empty then downloading data set addresses to the queue (column 3 line 1 – column 4 line 5).

It would have been obvious for one of ordinary skill in the art to download the data set corresponding to the representations in the hash table to the disk file if the disk file is empty as per the teaching of Najork so that new URLs can be stored as they are processed.

Monier discloses determining if the representations have been previously stored in the hash table/disk file (columns 8 & 9). Monier fails to teach when the representation is not stored in the buffer and the disk file is not empty, storing the representation in the buffer and delaying scheduling of the corresponding data set for downloading until it is determined that the representation has not previously stored in the disk file. However, Najork '265 teaches determining if the queue is not empty then delaying and assigning a download time for the data set addresses (column 3 line 1 – column 4 line 5).

> It would have been obvious for one ordinarily skilled in the art to assign a download time for the data set addresses as per the teachings of Najork '265 that would allow sufficient time to determine if the representations are stored in the disk file.
>
> Monier fails to teach performing an ordered merge of the contents of the buffer into the contents of the first disk file wherein the ordered merge comprises preventing duplication of any of the representations of data set addresses stored in the first disk file. However, Najork '614 teaches storing an index of representations and performing a sorted merge of the index with a disk file, preventing duplication (column 3 line 30 – column 4 line 45 and column 6 line 1 – column 7 line 25).
>
> It would have been obvious for one having ordinary skill in the art to perform a sorted merge of the hash table with the disk file as per the teachings of Najork '614 for preventing duplication and facilitating look-up operations on the disk file.

Office Action, pages 8-10.

The proposed combination of Monier, Najork '265, and Najork '614 cannot render the Applicant's claims obvious under Section 103 because Monier, Najork '265, and Najork '614 do not disclose all of the elements recited in the Applicant's claims. Independent claim 22 recites "performing an ordered merge ... preventing duplication of any of the representations of data set addresses stored in the first disk file."

As previously discussed above, Monier merely discloses appending the contents of a buffer to the end of a disk file. Najork '265 does not remedy this shortcoming of Monier because Najork '265 also fails to disclose an ordered merge preventing duplication of the merged contents. Najork '265 instead discloses a method for scheduling web crawlers to efficiently download web pages from the World Wide Web. Col. 1, lines 5-9. Nothing in Najork '265 suggests the desirability of the "ordered merge ... preventing duplication of any of the representations of data set addresses stored in the first disk file," as set forth in claim

22. Najork '614 also fails to disclose an ordered merge preventing duplication of the merged contents. Instead, Najork '614 discloses a processing procedure comprising three lookup steps and a merging process wherein the stored contents of a cache and disk file are combined and reorganized into a sorted order. Col. 6, line 1 – col. 7, line 25. When the Najork '614 procedure activates the merging process, the disk file has *already* been checked for duplicates. Unlike embodiments of the present invention which increase efficiency by performing an "ordered merge ... wherein the ordered merge *comprises* preventing duplication of any of the representations of data set addresses stored in the first disk file," Najork '614 discloses accessing the disk file multiple times to prevent duplication *before* merging the cache with the disk file. (Emphasis Added).

In view of the arguments presented above, the Applicant asserts that independent claim 22 is in condition for allowance. Accordingly, the Applicant requests that the Examiner withdraw the objection of claim 22 and provide an indication of allowance.

Conclusion

In view of the remarks set forth above, the Applicant respectfully requests withdrawal of all of the Examiner's rejections. Furthermore, the Applicant asserts that an indication of the allowability of claims 1-52 is appropriate. If the Examiner believes that a telephonic interview will help speed this application toward issuance, the Examiner is invited to contact the undersigned at the telephone number listed below.

Respectfully submitted,

Date: November 9, 2004

Barry D. Blount
Reg. No. 35,069
FLETCHER YODER
(281) 970-4545

CORRESPONDENCE ADDRESS:

Intellectual Property Administration
Legal Department, M/S 35
HEWLETT-PACKARD COMPANY
P.O. Box 272400
Fort Collins, CO 80527-2400